One Semi-Automated Forces (OneSAF)

Capabilities, architecture, and processes

John Logsdon, Deputy PM, OneSAF

Dave Nash, Ph.D.

Mike Barnes

# Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **10 MAR 2008**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **One Semi-Automated Forces (OneSAF) Capabilities, architecture, and processes**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **TRADOC Analysis Center**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
2008 DoD M&S (Modeling and Simulation) Conference Presentations held in Orlando, Florida on March 10 - 14, 2008, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**44**	

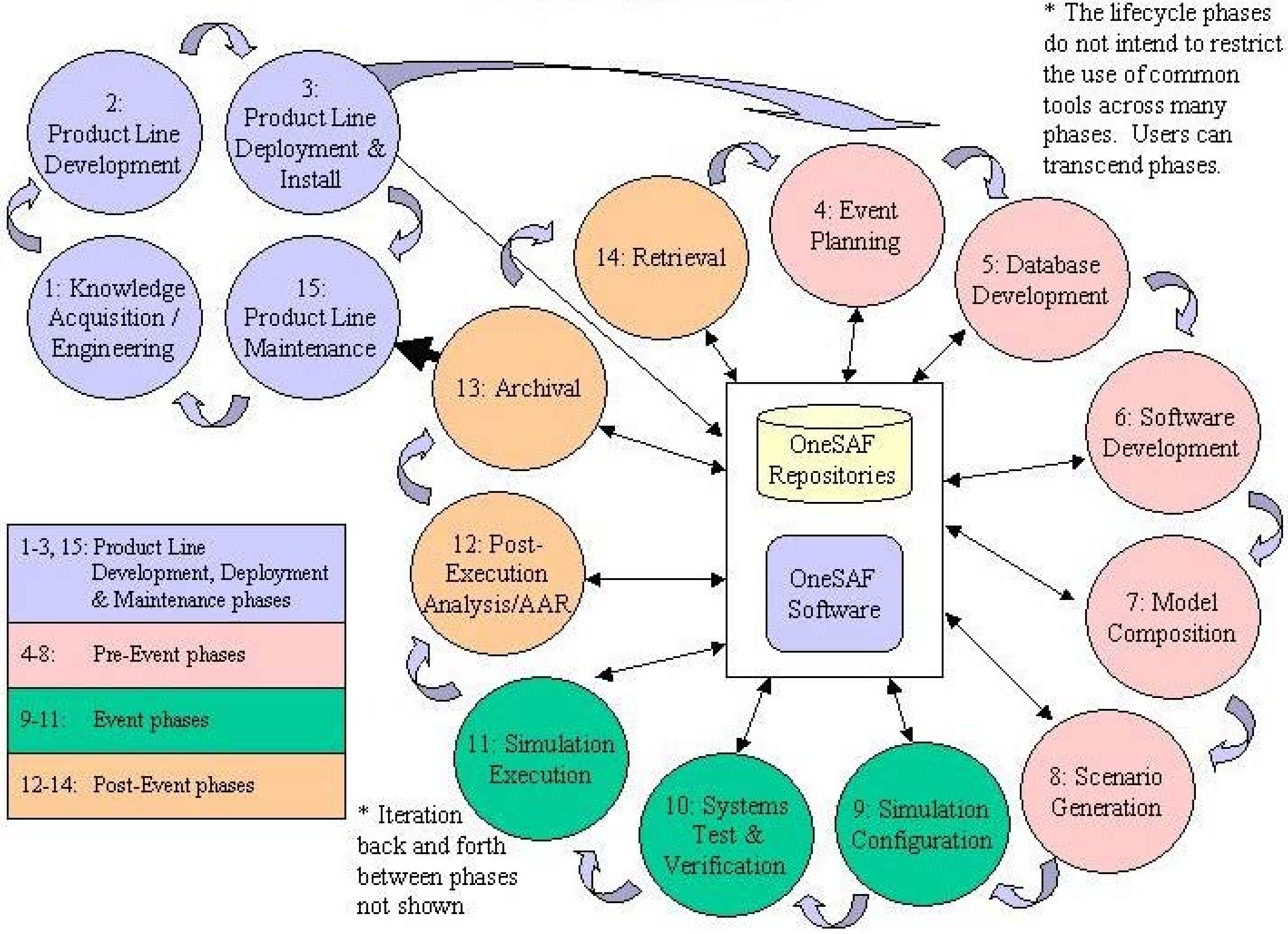
User communities

OneSAF is chartered to serve three communities, each of which has its own requirements and priorities:

- Advanced Concepts and Requirements (ACR)
- Research, Development and Acquisition (RDA)
- Training, Exercise and Military Operations (TEMO)

In addition to these, PM OneSAF also directs support that is to be provided to various joint and civilian organizations, and international customers.

The Simulation Lifecycle


2: Product Line Development
3: Product Line Deployment & Install
1: Knowledge Acquisition / Engineering
15: Product Line Maintenance
14: Retrieval
4: Event Planning
5: Database Development
13: Archival
OneSAF Repositories
OneSAF Software
6: Software Development
12: Post-Execution Analysis/AAR
7: Model Composition
11: Simulation Execution
10: Systems Test & Verification
9: Simulation Configuration
8: Scenario Generation

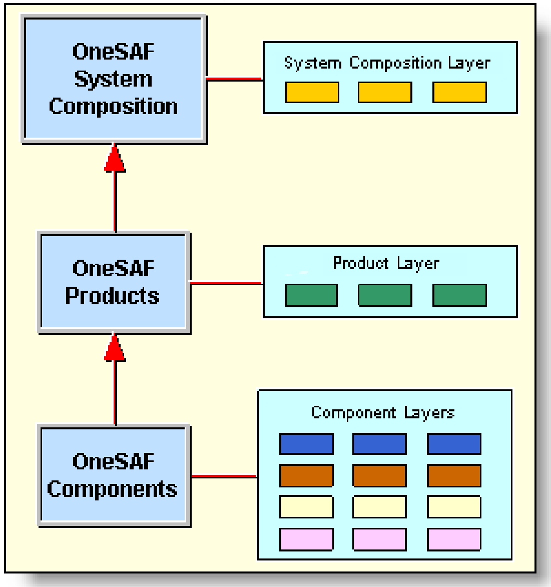
1-3, 15:	Product Line Development, Deployment & Maintenance phases
4-8:	Pre-Event phases
9-11:	Event phases
12-14:	Post-Event phases

* The lifecycle phases do not intend to restrict the use of common tools across many phases. Users can transcend phases.

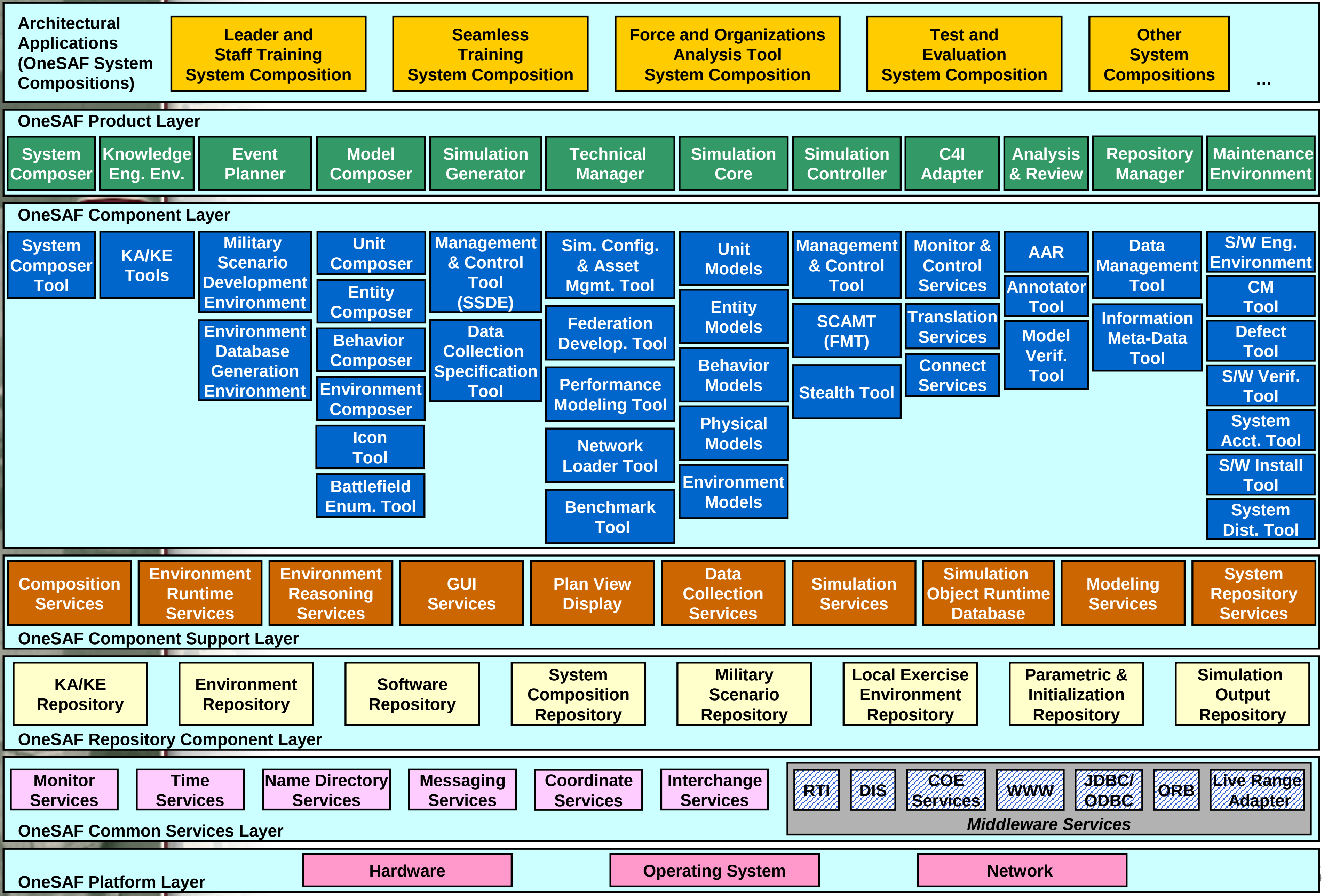
* Iteration back and forth between phases not shown

Composable Product Line Concept

- OneSAF is not a single system or product.
- OneSAF is a set of:
  - *Components*,
  - *Products* comprising OneSAF components,
  - *System compositions* comprising OneSAF products
- Tools are provided to create compositions of various types (including system compositions).



The Product Line Architecture Framework



Architectural Applications (OneSAF System Compositions)

- Leader and Staff Training System Composition
- Seamless Training System Composition
- Force and Organizations Analysis Tool System Composition
- Test and Evaluation System Composition
- Other System Compositions
- ...

OneSAF Product Layer

- System Composer
- Knowledge Eng. Env.
- Event Planner
- Model Composer
- Simulation Generator
- Technical Manager
- Simulation Core
- Simulation Controller
- C4I Adapter
- Analysis & Review
- Repository Manager
- Maintenance Environment

OneSAF Component Layer

- System Composer Tool
- KA/KE Tools
- Military Scenario Development Environment
- Environment Database Generation Environment
- Unit Composer
- Entity Composer
- Behavior Composer
- Environment Composer
- Icon Tool
- Battlefield Enum. Tool
- Management & Control Tool (SSDE)
- Data Collection Specification Tool
- Sim. Config. & Asset Mgmt. Tool
- Federation Develop. Tool
- Performance Modeling Tool
- Network Loader Tool
- Benchmark Tool
- Unit Models
- Entity Models
- Behavior Models
- Physical Models
- Environment Models
- Management & Control Tool
- SCAMT (FMT)
- Stealth Tool
- Monitor & Control Services
- Translation Services
- Connect Services
- AAR
- Annotator Tool
- Model Verif. Tool
- Data Management Tool
- Information Meta-Data Tool
- S/W Eng. Environment
- CM Tool
- Defect Tool
- S/W Verif. Tool
- System Acct. Tool
- S/W Install Tool
- System Dist. Tool

- Composition Services
- Environment Runtime Services
- Environment Reasoning Services
- GUI Services
- Plan View Display
- Data Collection Services
- Simulation Services
- Simulation Object Runtime Database
- Modeling Services
- System Repository Services

OneSAF Component Support Layer

- KA/KE Repository
- Environment Repository
- Software Repository
- System Composition Repository
- Military Scenario Repository
- Local Exercise Environment Repository
- Parametric & Initialization Repository
- Simulation Output Repository

OneSAF Repository Component Layer

- Monitor Services
- Time Services
- Name Directory Services
- Messaging Services
- Coordinate Services
- Interchange Services
- RTI
- DIS
- COE Services
- WWW
- JDBC/ ODBC
- ORB
- Live Range Adapter

Middleware Services

OneSAF Common Services Layer

- Hardware
- Operating System
- Network

OneSAF Platform Layer

MSDE / AAR

- Military Scenario Definition Environment (MSDE)
  - PowerPoint-based
  - Creates scenario definition (units, command relationships, dispositions) in Military Scenario Definition Language (MSDL)
  - MSDL is imported into a OneSAF scenario for execution
- After-action Review (AAR) Tool
  - Generates slides of the simulated scenario
  - Statistics of “snapshot” data items

Composability

- Composability:
  - The ability to assemble reusable software elements (“components”) into customized software.
  - A central design theme of OneSAF.

- Fairly recent capability in software development enabled by:
  - increased computing power,
  - success in developing reusable software libraries, and
  - the development and distribution of software and software components over the internet to different computer systems.

- Instead of developing complete computer programs for each computer system, software developers develop platform independent components to be assembled on the user’s computer.

Composition nomenclature

- Use of the term "composition" is ubiquitous in OneSAF, and usually requires context to know exactly what is being described:
  - *System composition*: a JAR file consisting of a set of system components
  - *Component composition*: an XML file containing pointers to various Java classes; used to imbue actors with capabilities such as:
    - Vulnerability, mobility, and sensing (physical components)
    - Responsiveness to orders, ability to process fire missions (behavioral components)
  - Actor compositions:
    - *Entity composition*: an XML file containing metadata and pointers to component compositions defining a single, indivisible battlespace object
    - *Unit composition*: an XML file containing metadata and pointers to component compositions defining a single, indivisible battlespace object
  - *Behavior composition*: an XML file containing metadata, inputs, (in the case of composite behaviors) outputs, and control definitions defining the effect of a behavior

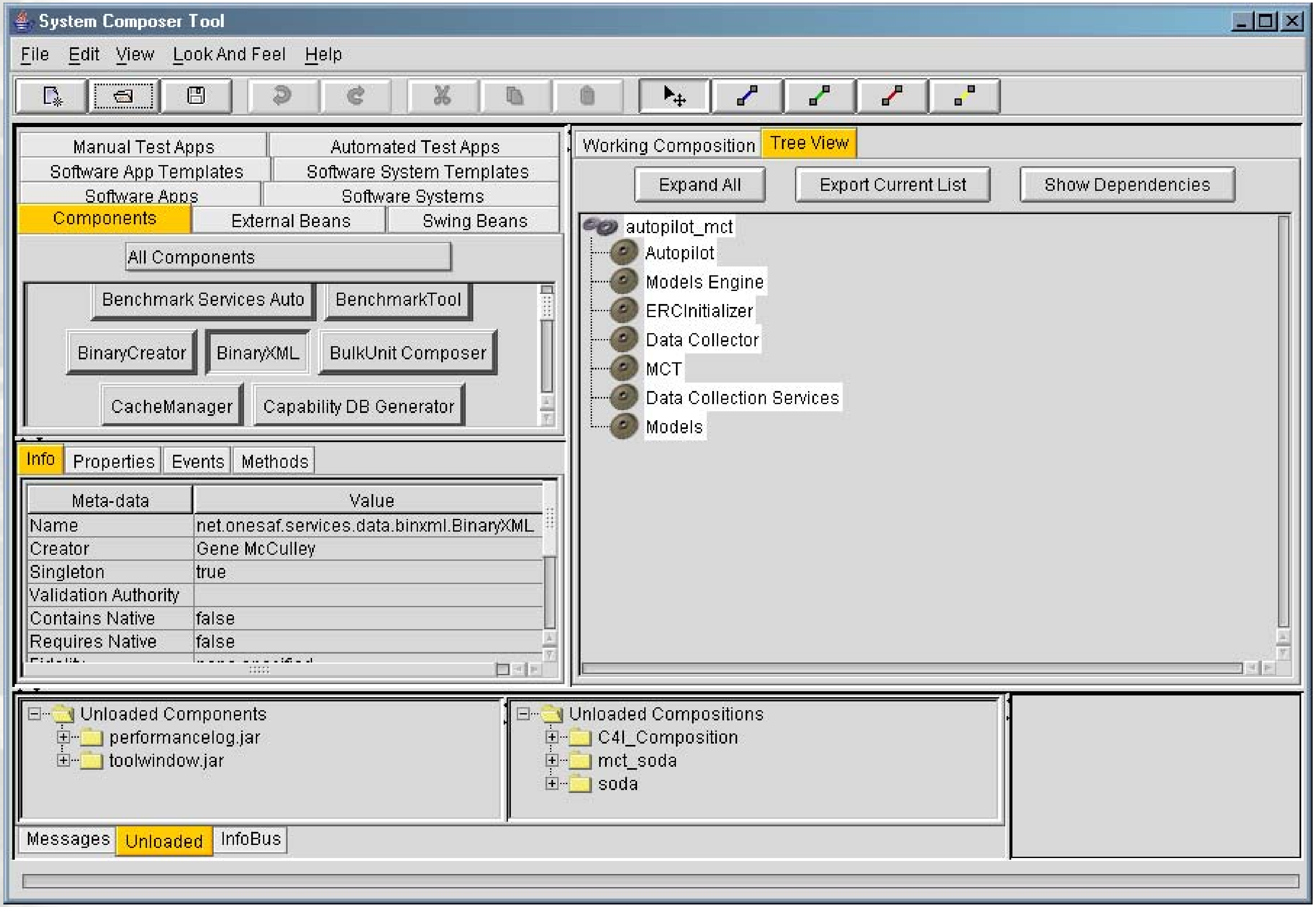
OneSAF System Components

- A OneSAF *system composition* is an assembly of system components. In some ways a system composition is analogous to an executable program.
- A *system component* is a modular software artifact that makes specific functionality available to the system composition.
- Some examples:
  - The Formation Editor is a system component providing a graphical tool that may be used to interactively design formation data files.
  - The formationEditor system composition consists of the Formation Editor and the Desktop components.
- OneSAF components are JavaBeans with extensions; e.g.:
  - OneSAF specific component metadata such as releasability, dependencies, and fidelity
  - Non-Java code support to allow reuse of code written in other languages.
- Components can contain any subset of:
  - Data
  - Software source code
  - Software executable code
  - Documentation

System Composer

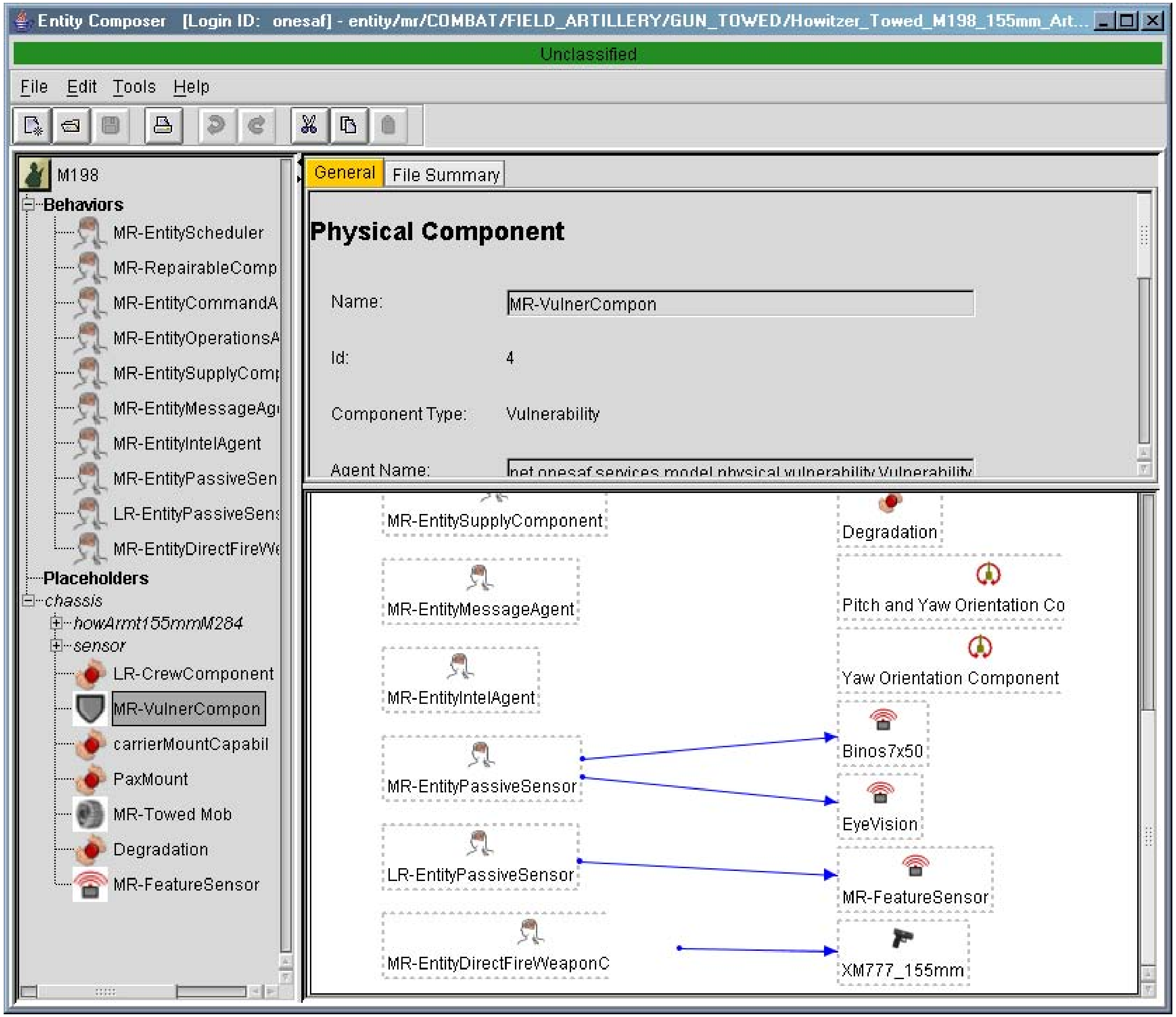
- The System Composer is a graphical tool used to create OneSAF system compositions:
  - The System Composer is based on JavaBean application builders but works with OneSAF extended JavaBeans (OneSAF system components).
  - The System Composer creates a JAR file for each composition (the composition and JAR file have the same name) containing:
  - Information the runtimeloader script uses to run the composition.
  - Composition metadata; e.g.,
    - Description
    - Specification and Implementation Version
    - Validation Authority and Date
  - The System Composer provides the software “glue” connecting components.
- It is executed from the command prompt by running the composer script within SWR/bin/services/sys/composition
  - This is the only stand-alone application within OneSAF; all other applications are started using the runtimeloader script in the same directory

System Composer



Meta-data	Value
Name	net.onesaf.services.data.binxml.BinaryXML
Creator	Gene McCulley
Singleton	true
Validation Authority	
Contains Native	false
Requires Native	false

Entity Composer



Entity Composer [Login ID: onesaf] - entity/mr/COMBAT/FIELD_ARTILLERY/GUN_TOWED/Howitzer_Towed_M198_155mm_Art...

Unclassified

Physical Component

Name: MR-VulnerCompon

Id: 4

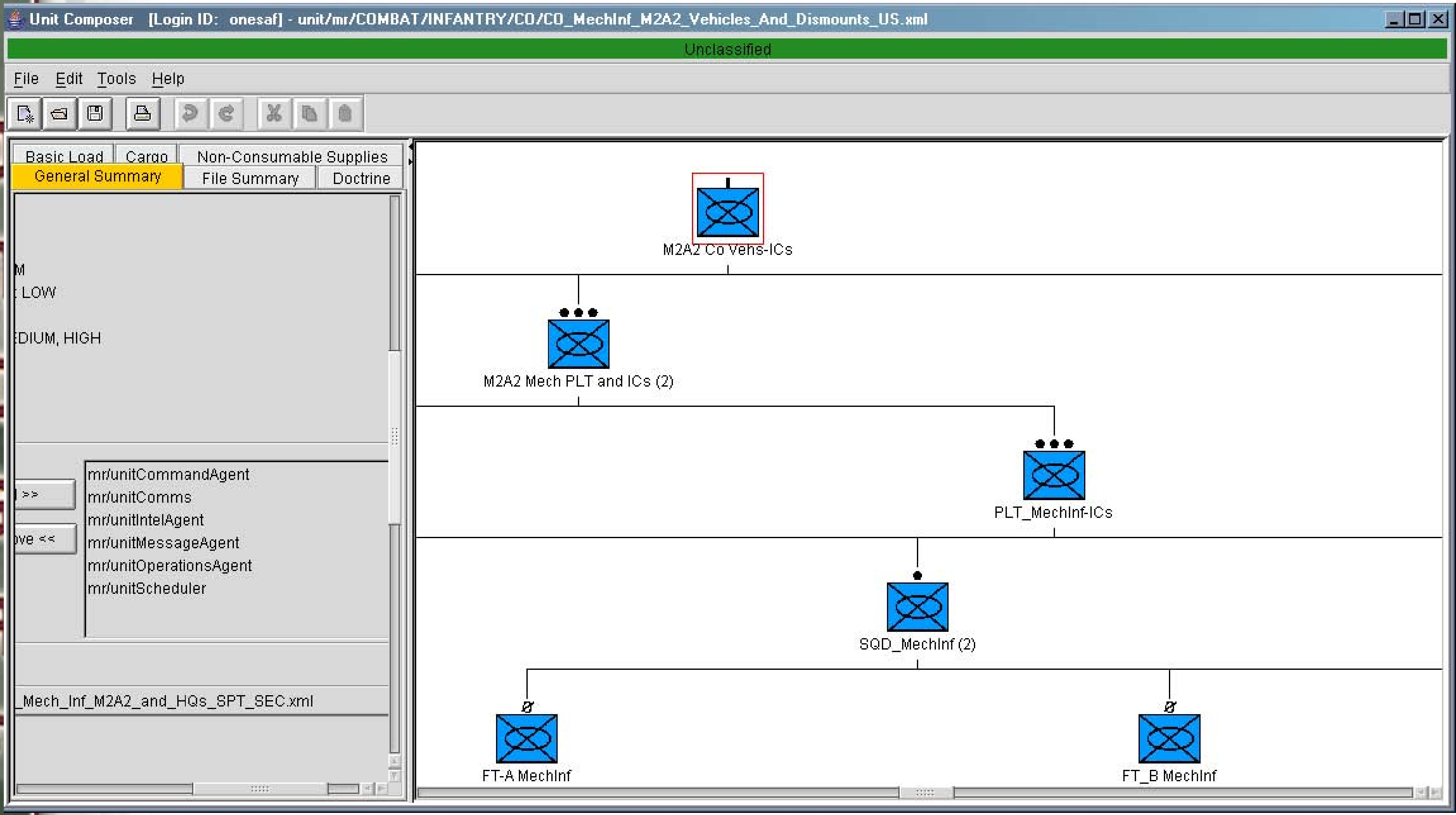
Component Type: Vulnerability

Agent Name: net.onesaf.services.model.physical.vulnerability.Vulnerability

Using the Entity Composer

- System composition = `composers`, therefore `$BINDIR/runtimeloader composers`
- Typical components
  - Physical
    - Sensors (eyeballs, FLIR, etc.)
    - Weapons
    - Mobility
    - Vulnerability
  - Behavior
    - Direct fire controller
    - Operations
    - Intel
    - Supply

Unit Composer



M

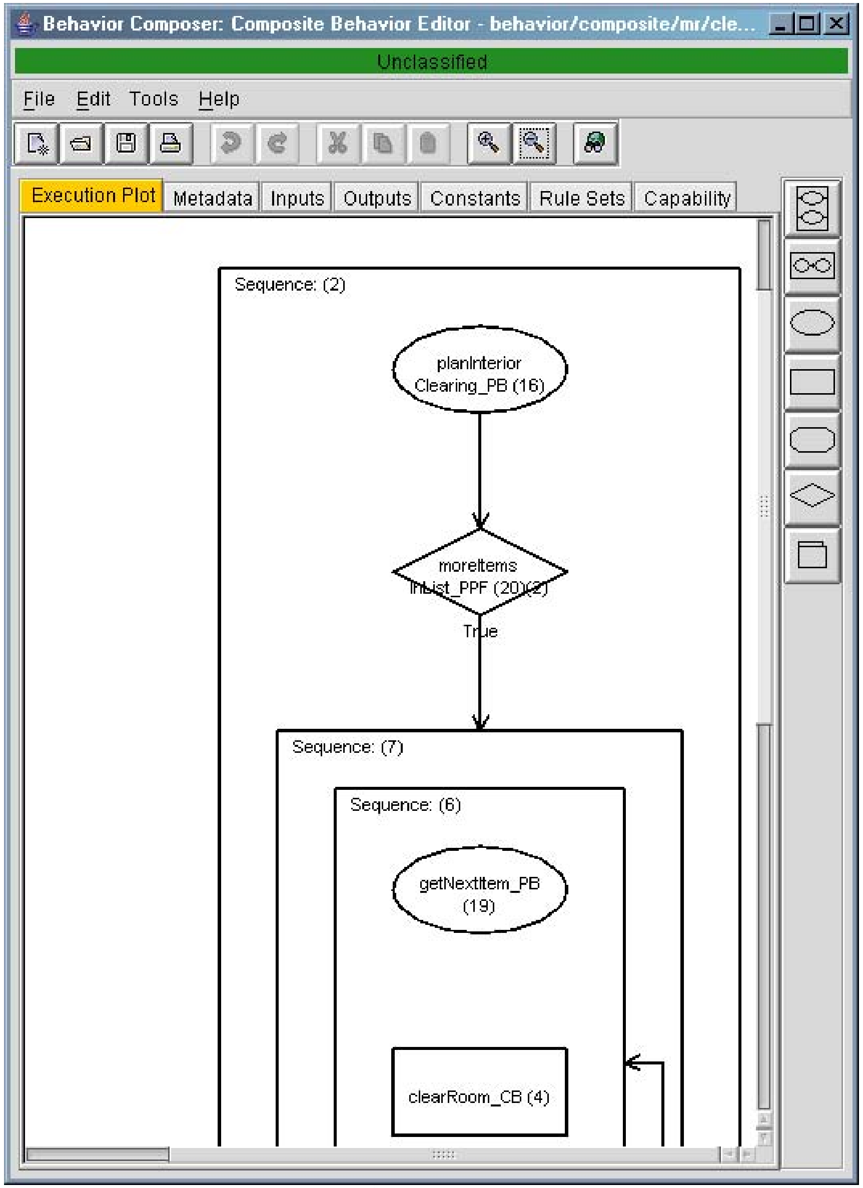
LOW

DIUM, HIGH

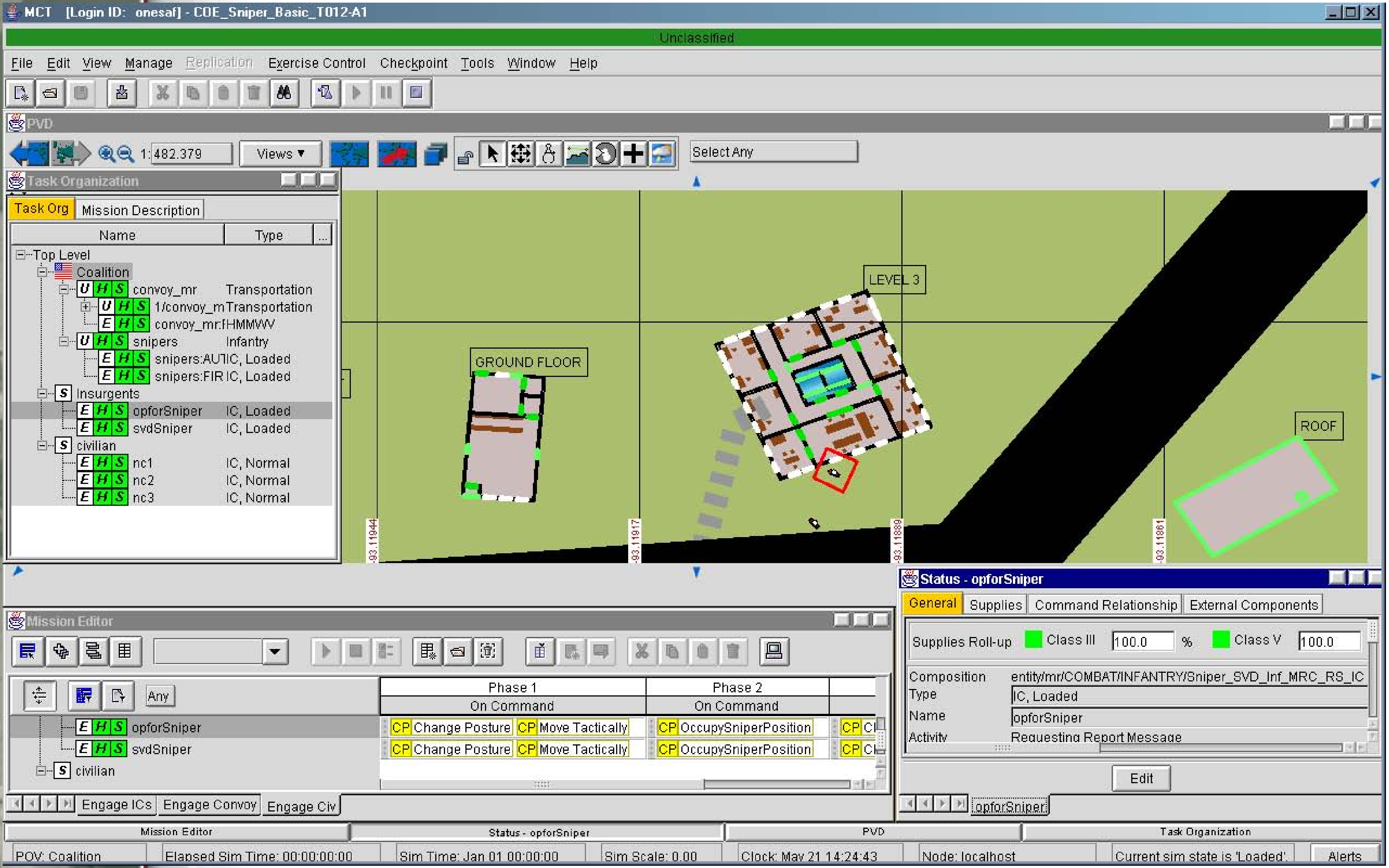
mr/unitCommandAgent
mr/unitComms
mr/unitIntelAgent
mr/unitMessageAgent
mr/unitOperationsAgent
mr/unitScheduler

_Mech_Inf_M2A2_and_HQs_SPT_SEC.xml

Behavior Composer



The Management and Control Tool



MCT [Login ID: onesaf] - COE_Sniper_Basic_T012-A1

Unclassified

Name	Type
Top Level	
Coalition	
U H S convoy_mr	Transportation
U H S 1/convoy_m	Transportation
E H S convoy_mr:f	HMMWV
U H S snipers	Infantry
E H S snipers:AUT	IC, Loaded
E H S snipers:FIR	IC, Loaded
S Insurgents	
E H S opforSniper	IC, Loaded
E H S svdSniper	IC, Loaded
S civilian	
E H S nc1	IC, Normal
E H S nc2	IC, Normal
E H S nc3	IC, Normal

	Phase 1	Phase 2	
	On Command	On Command	
E H S opforSniper	CP Change Posture CP Move Tactically	CP OccupySniperPosition	CP Cl
E H S svdSniper	CP Change Posture CP Move Tactically	CP OccupySniperPosition	CP Cl
S civilian			

Supplies Roll-up 🟩 Class III 100.0 % 🟩 Class V 100.0

Composition entity/mr/COMBAT/INFANTRY/Sniper_SVD_Inf_MRC_RS_IC

Type IC, Loaded

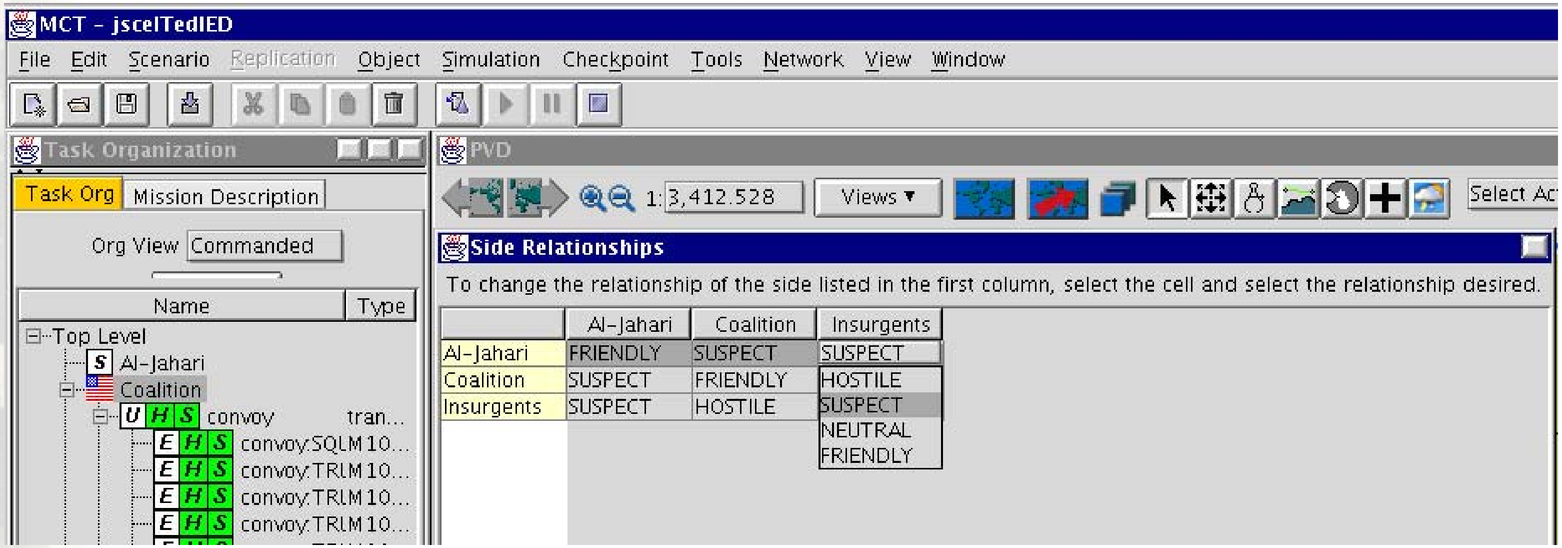
Name opforSniper

Activity Requesting Report Message

POV: Coalition | Elapsed Sim Time: 00:00:00:00 | Sim Time: Jan 01 00:00:00 | Sim Scale: 0.00 | Clock: May 21 14:24:43 | Node: localhost | Current sim state is 'Loaded'. | Alerts

Sides and Forces

- A *force* is a collection of entities, units, or forces.
- A *side* is a collection of entities, units, forces, or sides.
- Sides have relationships which determine certain behaviors.



Org View Commanded

1: 3,412.528

To change the relationship of the side listed in the first column, select the cell and select the relationship desired.

	Al-Jahari	Coalition	Insurgents
Al-Jahari	FRIENDLY	SUSPECT	SUSPECT
Coalition	SUSPECT	FRIENDLY	HOSTILE
Insurgents	SUSPECT	HOSTILE	SUSPECT

Variable fidelity

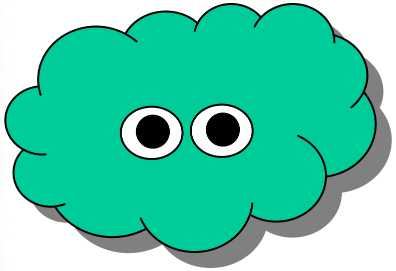
- Support for variable levels of fidelity makes it possible to tailor the simulation in order to maximize satisfaction of diverse use cases.

- Accurately and effectively represents activities within the Army warfighting functions:

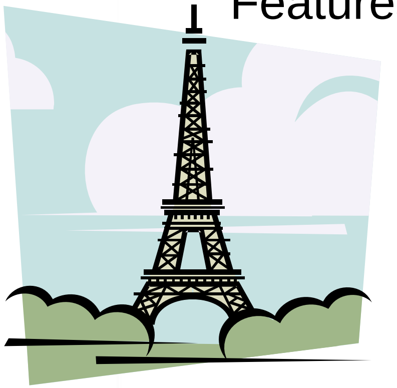
    - Intelligence
    - Movement and maneuver
    - Fire support
    - Protection
    - Sustainment
    - Command and control

Environmental Runtime Component (ERC)

Obscurants / contaminants

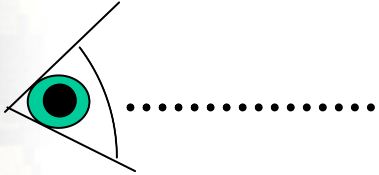


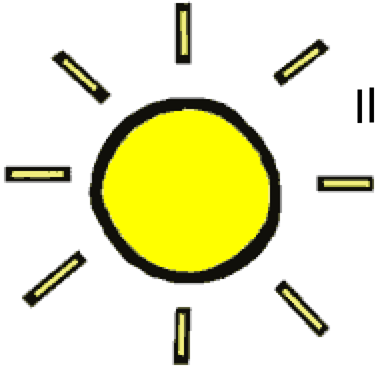
Features



Weather

Line of Sight





Illumination

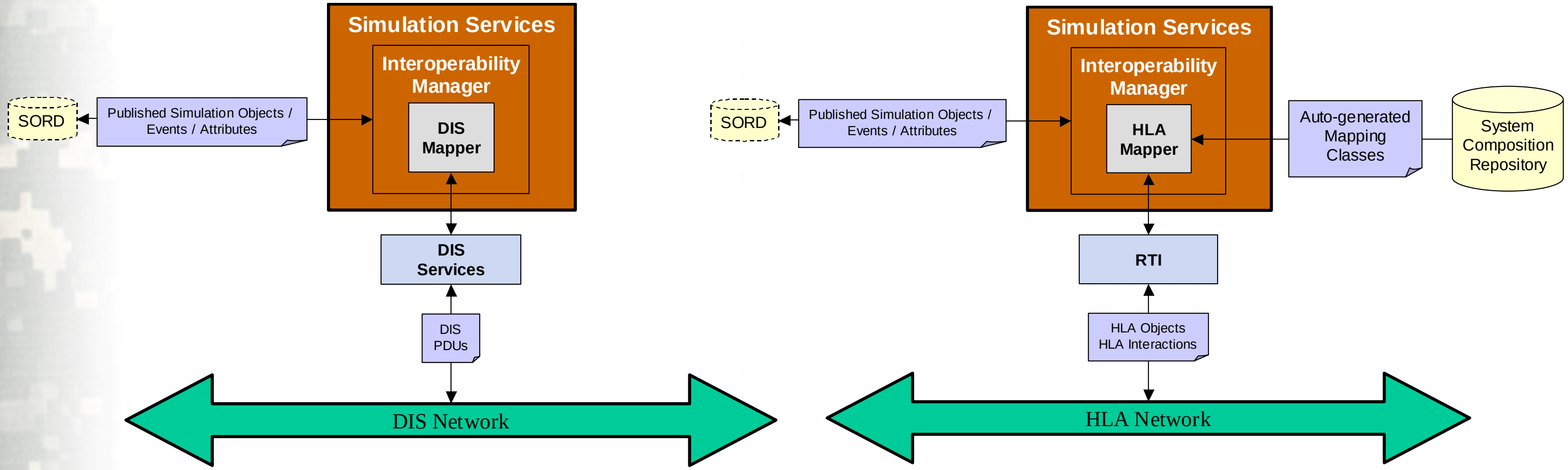
Terrain

Distributed Simulation

- OneSAF is a distributed simulation system.
- Workstations may be assigned different computational roles, such as:
  - Simulation core (SimCore)
  - Management and Control Tool interface
  - C4I adapter
- Distributed design allows:
  - Individual workstations to balance loads
  - System recovery of individual workstations without interrupting a simulation in progress

Interoperability

- OneSAF instances interoperate via:
  - Internal protocols
    - OneSAF messaging services
    - C4I Adapter
  - External protocols
    - HLA
    - DIS



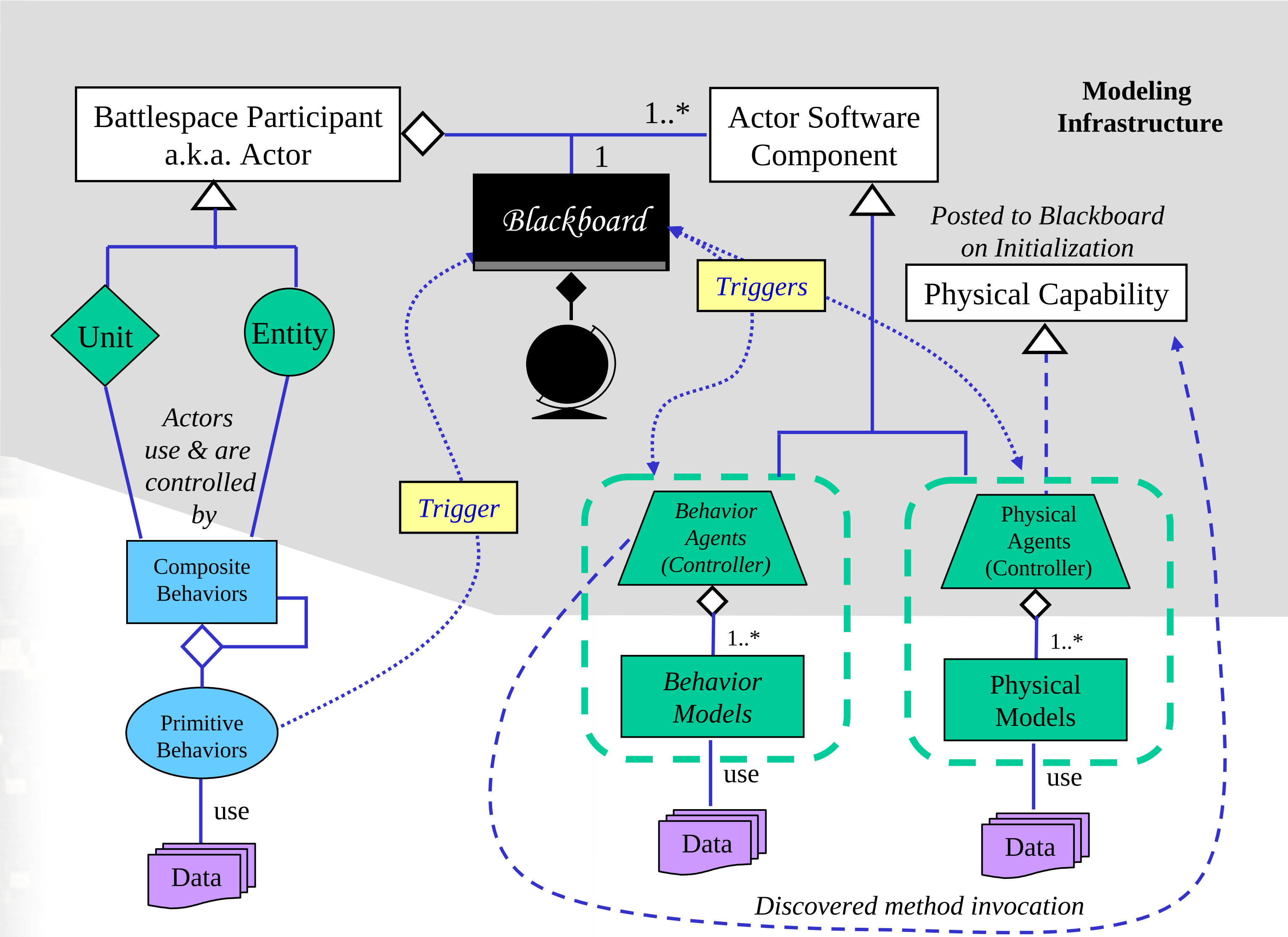
Simulation Services
Interoperability Manager
DIS Mapper
SORD
Published Simulation Objects / Events / Attributes
DIS Services
DIS PDUs
DIS Network
Simulation Services
Interoperability Manager
HLA Mapper
SORD
Published Simulation Objects / Events / Attributes
Auto-generated Mapping Classes
System Composition Repository
RTI
HLA Objects HLA Interactions
HLA Network

Overview Summary

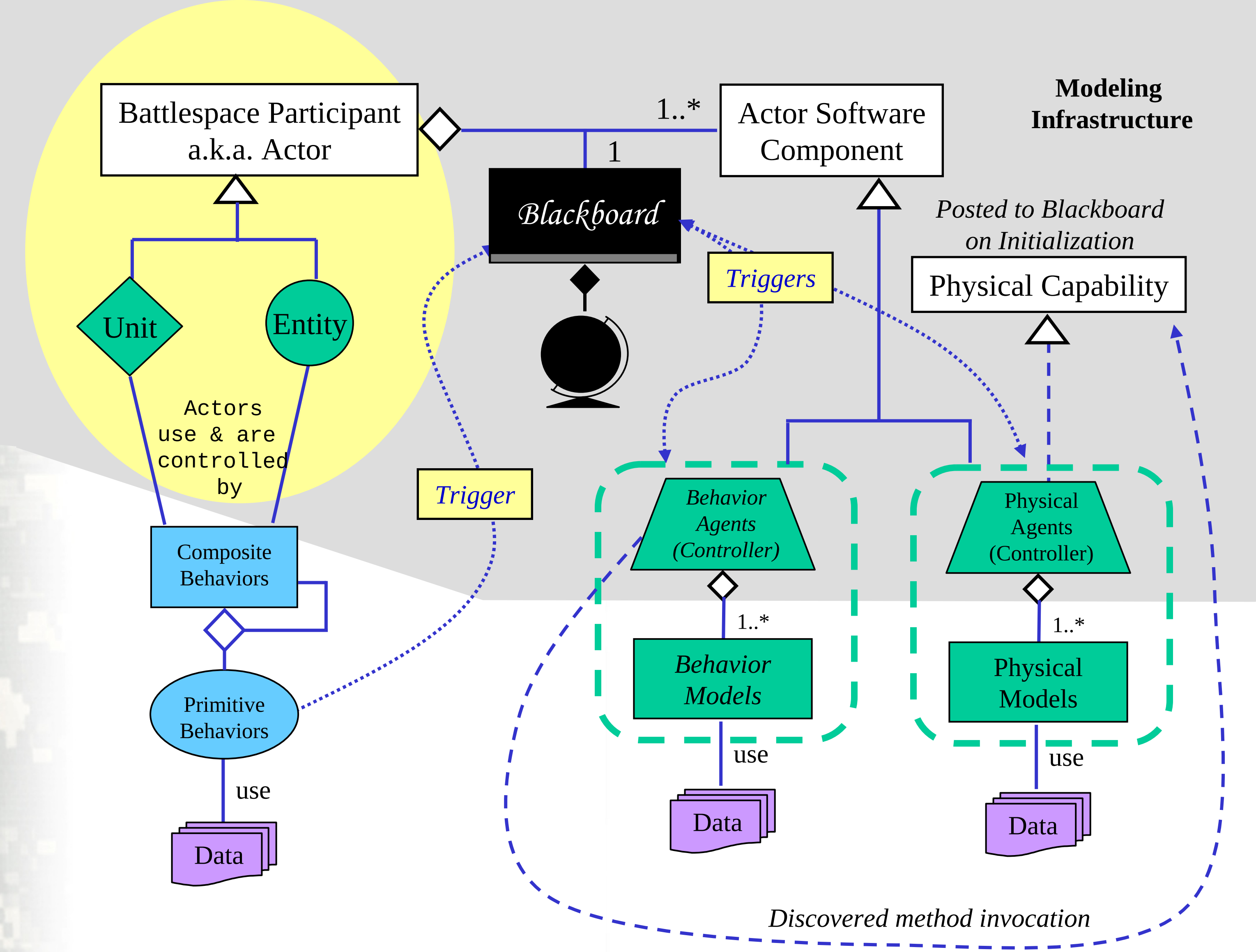
- OneSAF is a highly configurable suite of systems for the generation of semi-autonomous forces.
- Simulation behaviors and models are data driven to the maximum extent possible, to support customization and “what-if” analysis
- System capabilities are exposed by powerful visual tools.
- High degree of interoperability through DIS and HLA, and usage of common formats including XML for state.

Introduction to the OneSAF architecture

The OneSAF Agent Architecture



Battlespace participants (actors)



Battlespace Participant a.k.a. Actor

1..*

1

Actor Software Component

Modeling Infrastructure

Blackboard

Posted to Blackboard on Initialization

Triggers

Physical Capability

Unit

Entity

Actors use & are controlled by

Trigger

Composite Behaviors

Primitive Behaviors

use

Data

Behavior Agents (Controller)

1..*

Behavior Models

use

Data

Physical Agents (Controller)

1..*

Physical Models

use

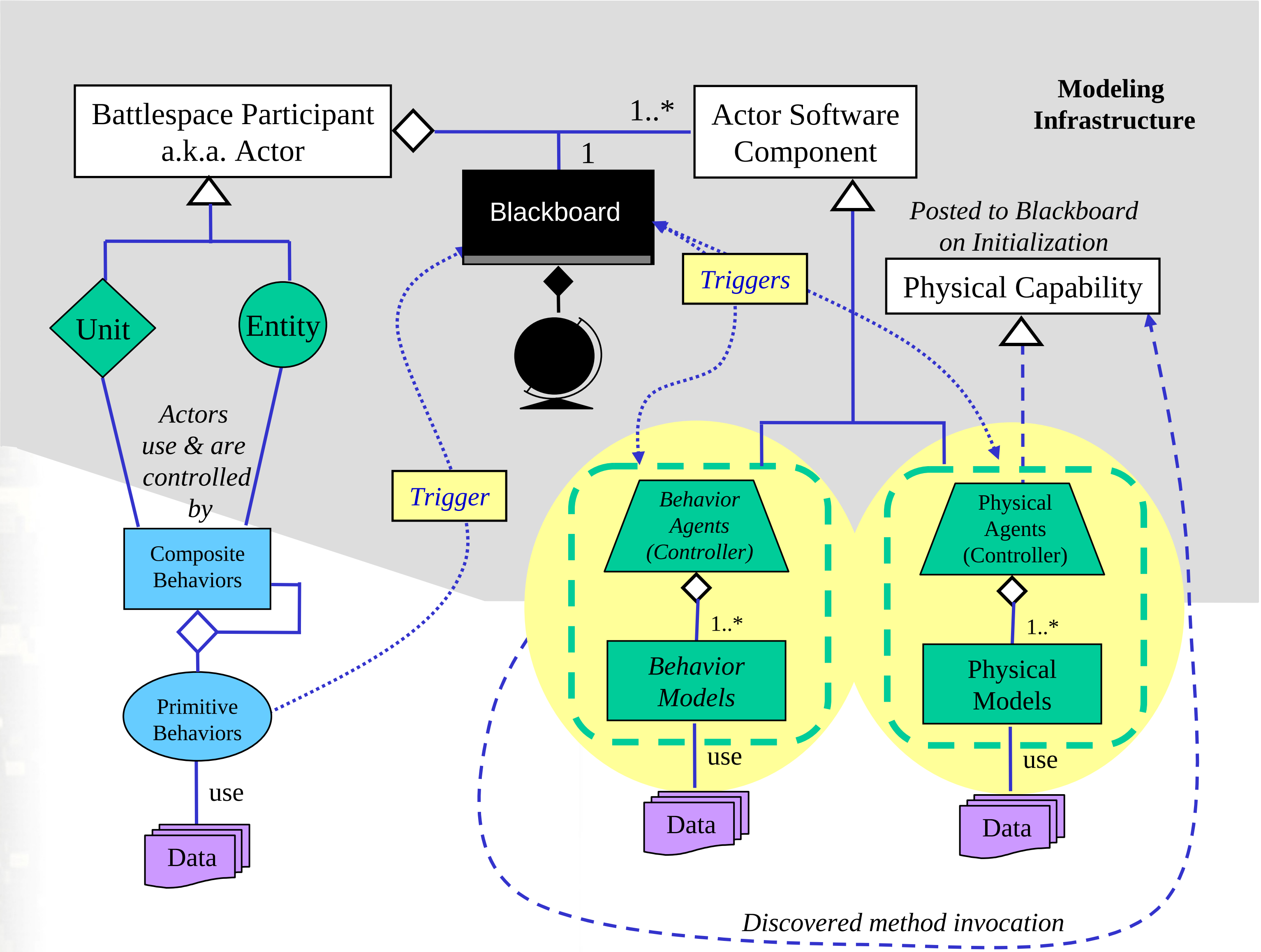
Data

Discovered method invocation

Battlespace participants (actors)

- Actor:
  - An entity or unit
  - A simulated thing that can be instantiated in a battlespace and has a location
- Entity:
  - The smallest discrete, stand-alone actor.
  - Implemented as a composition of physical components and behavioral components.
- Unit:
  - An organized collection of actors and their capabilities.
    - Alternatively … “a collection of actors”
  - A simulation object representing the combined command and control of a collection of subordinate actors (entities and sub-units).
    - Alternatively … “a collection of components”

Components



Modeling Infrastructure

Battlespace Participant a.k.a. Actor

1..*

1

Actor Software Component

Blackboard

Posted to Blackboard on Initialization

Physical Capability

Triggers

Unit

Entity

Actors use & are controlled by

Trigger

Composite Behaviors

Primitive Behaviors

use

Data

Behavior Agents (Controller)

1..*

Behavior Models

use

Data

Physical Agents (Controller)

1..*

Physical Models

use

Data

Discovered method invocation

Components

- Agents:
  - Behavioral agents provide command and control capabilities, such as planning, plan execution, and situation assessment.
  - Physical agents are the “middlemen” between behaviors, the physical world, and physical models.

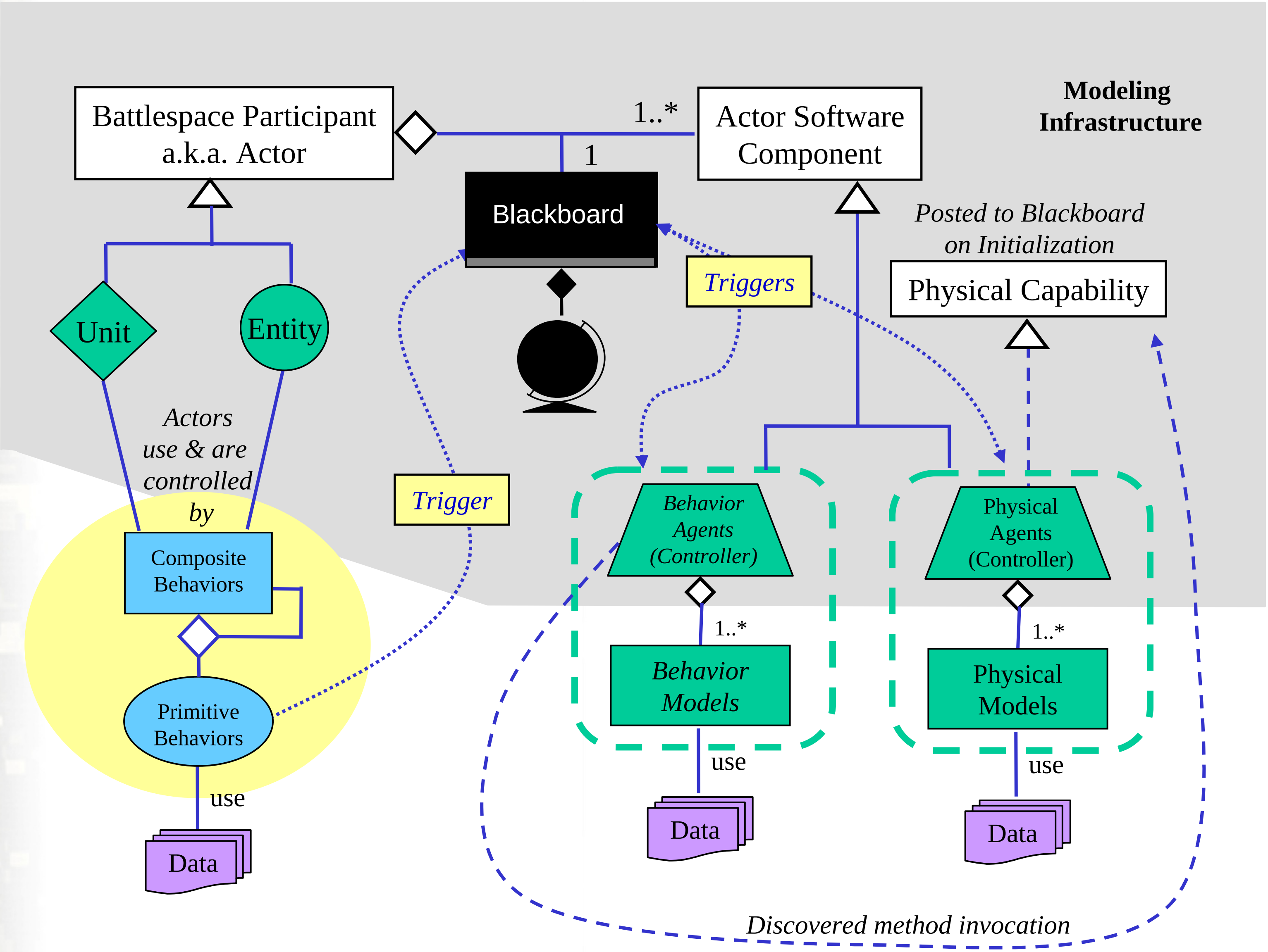
- Models:
  - Behavioral models answer behavior agents’ questions and represent the reasoning of agents.
  - Physical models provide physical capabilities, such as mobility, weapons, vulnerability, sensing, and communications. They represent the effectors and perceptors of simulated platforms and the physics of the simulated world.

Example Agents

- **Behavioral Agents:**
  - Command
  - Scheduler
  - Intel
  - Message
  - Operations
  - Driver
  - Direct Fire Weapons Controller
  - Fire Direction Center
  - Caller For Fire
  - ADA Target Handoff

- **Physical Agents:**
  - Weapon
  - Radio
  - Sensor
  - Mobility
  - Transport
  - Vulnerability

Behaviors


Modeling Infrastructure
Battlespace Participant a.k.a. Actor
1..*
1
Actor Software Component
Blackboard
Posted to Blackboard on Initialization
Triggers
Physical Capability
Unit
Entity
Actors use & are controlled by
Trigger
Composite Behaviors
Behavior Agents (Controller)
Physical Agents (Controller)
1..*
1..*
Behavior Models
Physical Models
Primitive Behaviors
use
use
use
Data
Data
Data
Discovered method invocation

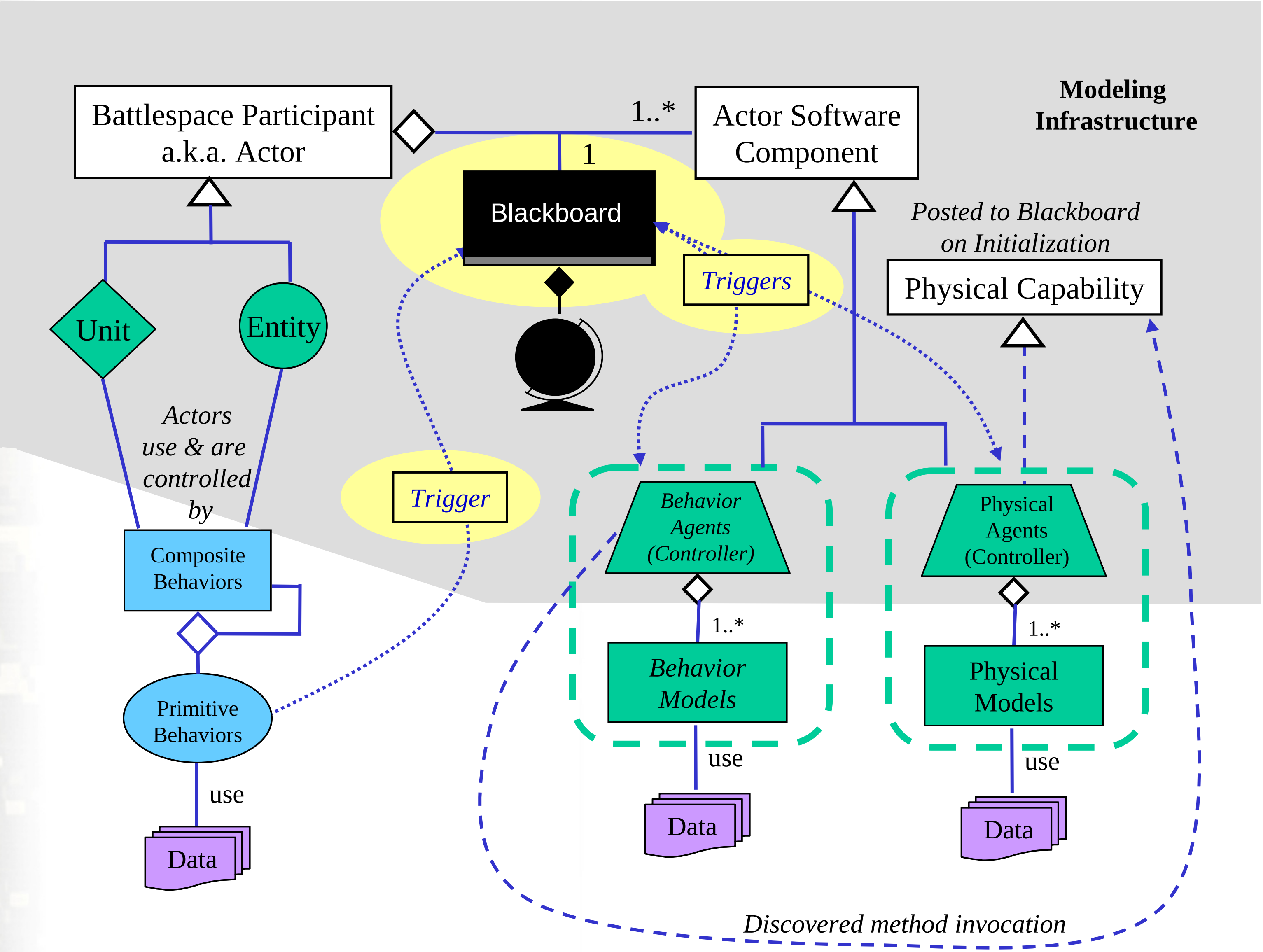
Composite behaviors

- Composite Behaviors:
  - Represent tasks and missions
  - Are composed of primitive and other composite behaviors
  - Are created with the Behavior Composer.
- Temporal organization:
  - Sequential
  - Parallel
- Conditional Branching
- Looping

Primitive behaviors

- Primitive behaviors:
  - Provide chunks of functionality from which more complex behavior models are built.
  - Are parameterized with inputs, and may have outputs
  - Interact with behavioral agents.
- Are implemented as Java classes, with an XML wrapper to allow composition

State transitions



Content carriers

- Event
  - Information object that is sent from an actor to the external world over the network.
  - Events are “sent” on the EventRouter.
  - Also referred to as a “Simulation Event.”
- Trigger
  - Information object sent internal to an actor. May be sent/received by agents and behaviors.
  - Triggers are “posted” on the blackboard.
  - Also referred to as a “Command.”
- Directive
  - A special event that can also function as a trigger.
  - The Directive may be “sent” or “posted.” When it is received as an event, it will automatically be posted as a trigger on the blackboard.
  - Also referred to as an “Intervention” (on the MCT).
- Fact
  - Information object that is stored in the blackboard.
  - Facts may be simple “Information Facts” that store information or they may be “Fuzzy Facts” that can evaluate its stored information and modify it over time.
- Message
  - A class of events used by the communication framework.

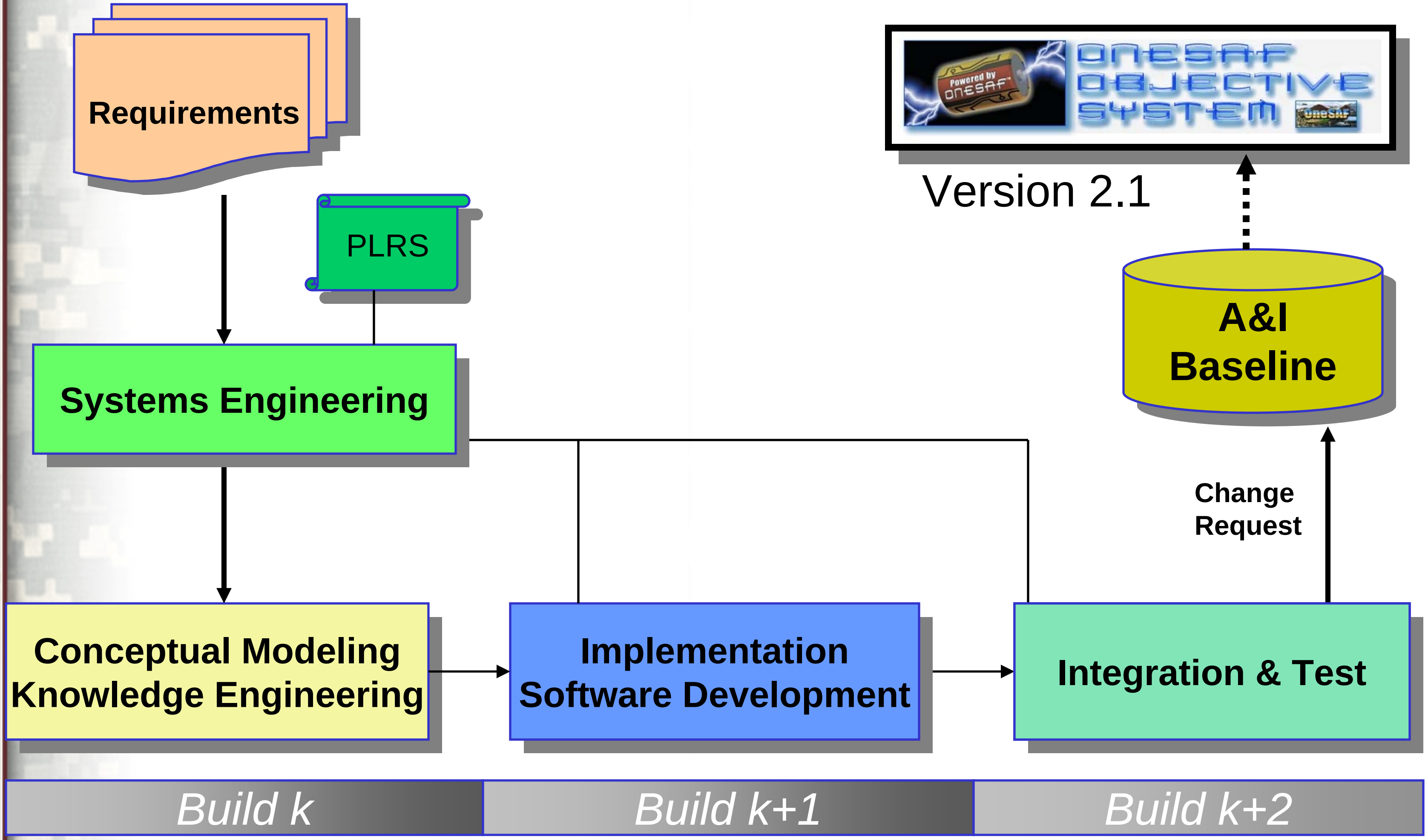
The Blackboard

- The Blackboard is an Agent control mechanism:
  - Agents subscribe to triggers on the blackboard.
  - When the blackboard receives a trigger, the it notifies that trigger’s subscribers (and only those subscribers).

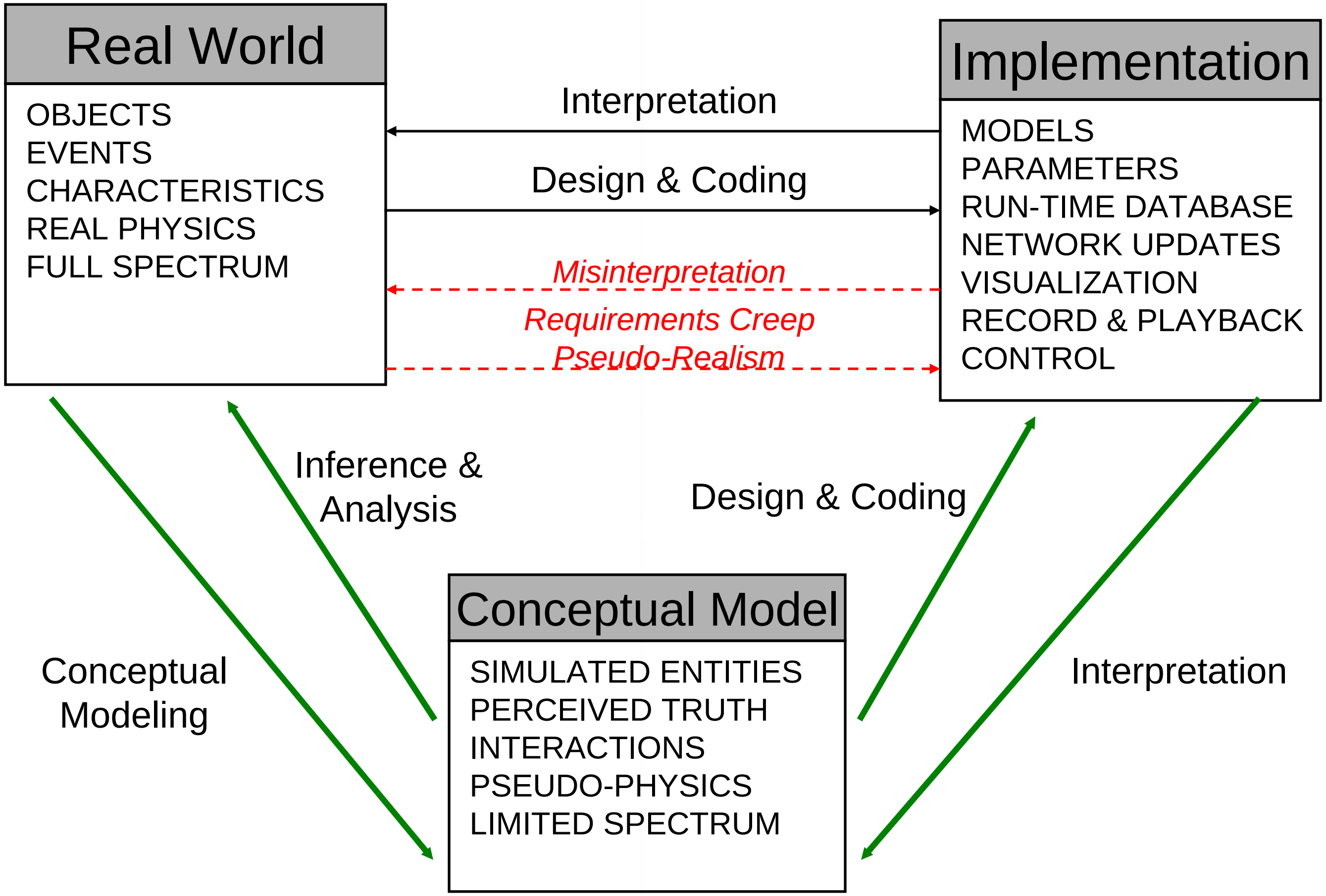
- Blackboard:
  - “World Model” for an actor containing perceived truth.
    - » Send and receive (internal) triggers
    - » Contains facts
    - » Access physical capabilities
  - One instance per actor; cannot be shared with other actors.

OneSAF processes

Capability Development Process Flow



What problem does Conceptual Modeling address?



Real World

OBJECTS
EVENTS
CHARACTERISTICS
REAL PHYSICS
FULL SPECTRUM

Interpretation

Design & Coding

Misinterpretation

Requirements Creep

Pseudo-Realism

Implementation

MODELS
PARAMETERS
RUN-TIME DATABASE
NETWORK UPDATES
VISUALIZATION
RECORD & PLAYBACK
CONTROL

Inference & Analysis

Design & Coding

Conceptual Modeling

Conceptual Model

SIMULATED ENTITIES
PERCEIVED TRUTH
INTERACTIONS
PSEUDO-PHYSICS
LIMITED SPECTRUM

Interpretation

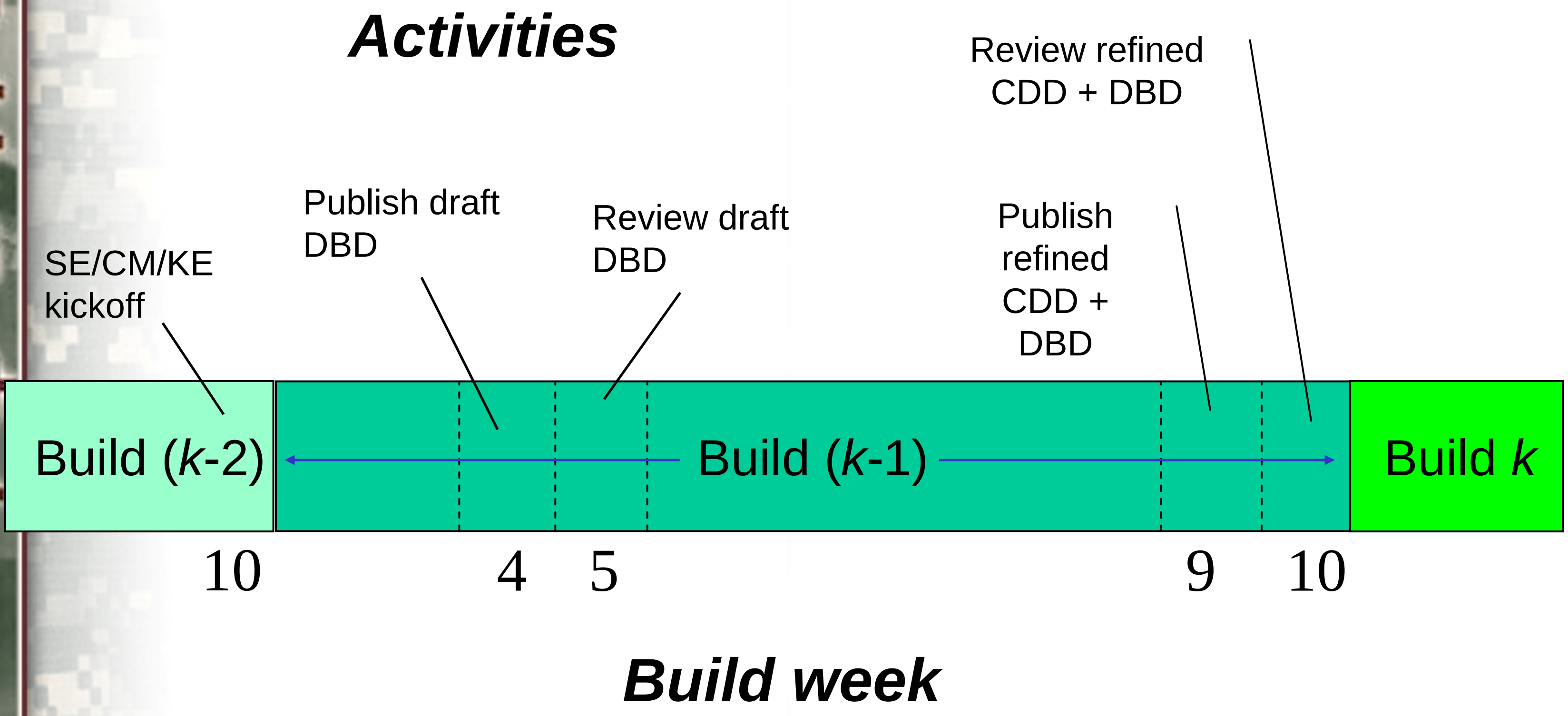
Simulating the real world is difficult and error prone.

A well-conceived, consistent intermediate model eliminates many problems by providing a model of the battlespace usable by all participants (customer, domain expert, developer, and user).

Conceptual Modeling and Knowledge Engineering Process

Main products:

- Capability Description Document (CDD)
- Domain Behavior Description (DBD)
- Physical Knowledge Acquisition Document (PKAD)



Activities

SE/CM/KE kickoff

Publish draft DBD

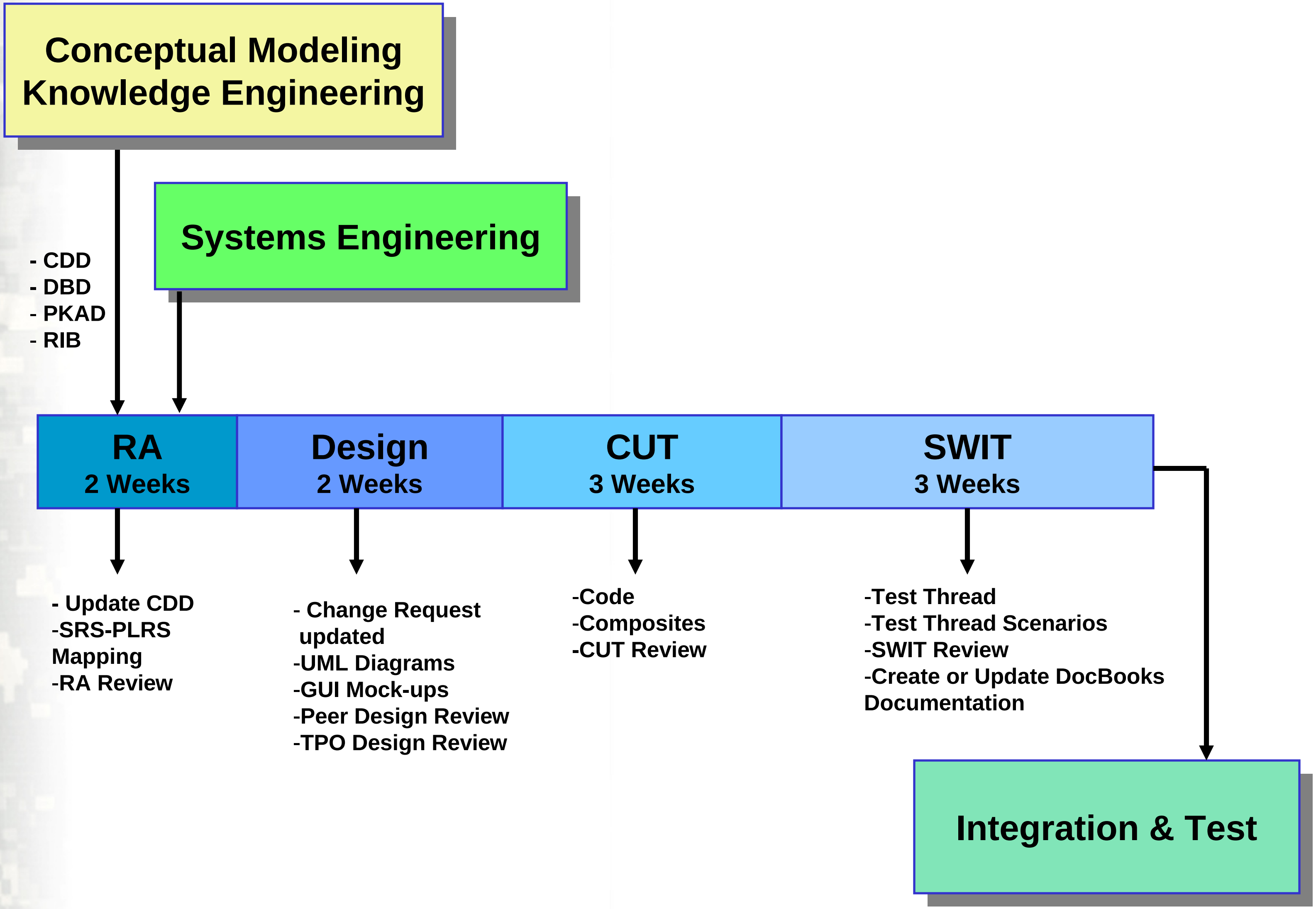
Review draft DBD

Publish refined CDD + DBD

Review refined CDD + DBD

Build week

Implementation Build



Conceptual Modeling Knowledge Engineering

- CDD
- DBD
- PKAD
- RIB

Systems Engineering

RA
2 Weeks

Design
2 Weeks

CUT
3 Weeks

SWIT
3 Weeks

- Update CDD
-SRS-PLRS Mapping
-RA Review

- Change Request updated
-UML Diagrams
-GUI Mock-ups
-Peer Design Review
-TPO Design Review

-Code
-Composites
-CUT Review

-Test Thread
-Test Thread Scenarios
-SWIT Review
-Create or Update DocBooks Documentation

Integration & Test

Co-developer handover process

General handover criteria

- *Integration with OneSAF*: It is expected that handovers are submitted for the purpose of integration into OneSAF, and that they are ready for integration.
- *Development baseline*: Capabilities must be developed against the most recent generally available release.
- *Architectural compliance*: Handover products must comply with architectural and design guidelines, protocols, language compatibilities, code structures, and documentation standards.
- *Testing*: Capabilities must be tested during development to ensure that they operate as designed are not injurious OneSAF.
- *Classification*: At present, all handover products must be unclassified.

Co-developer handover process

- To be considered for inclusion with a baseline release, co-developers participate in several activities:
  - Registration (co-developer)
  - Initial review (OneSAF Architecture and Integration)
  - Handover package submission (co-developer)
  - Engineering Change Control Board review
  - Peer review (OneSAF Architecture and Integration)
  - Baseline integration and test (OneSAF Architecture and Integration, with co-developer participation desired)
- Required artifacts:
  - Source code and data
  - Design documentation (use cases, sequence and class diagrams)
  - Domain documentation (PKAD / DBD / CDD)
  - Usage instructions
  - Unit tests and system test cases
  - Architectural compliance assessment

Questions?